Critical Software S.A.

Manual de Instalação do SISMA

Sistema de Informação de Saúde para Monitoria e Avaliação

REFERÊNCIA CONTRATO:
CSW-2012-PRL-02976

Código Projeto:	SISMA
Ref. Doc.:	CSWMZ-SISMA-2014-MAN-
Data:	2014-01-27
Páginas:	11
Estado:	Aprovado
Acesso:	Público
Versão:	02

SALVAGUARDA - Documento sob contrato com Jembi .

O trabalho descrito no presente documento foi desenvolvido sob contrato com Jembi. A responsabilidade pelo seu conteúdo é do autor ou organização que o preparou.

Parceiros:

Aprovação				
Versão	**Nome**	**Funções**	**Assinatura**	**Data**
01	Luis Gaspar	Gestor de Projecto		2014-01-27
01	Paulo Grácio	Gestor Técnico		2014-01-27

Autores e Contribuintes			
Nome	**Contacto**	**Descrição**	**Data**
Gulam Khan	gm-khan@criticalsoftware.com	Autor	2014-01-27

Lista de Acesso
Acesso Interno
Dep. Qualidade, Equipa de Projeto
Accesso Externo
Equipa de projecto, Dr. Alessandro Campione, Mr. Oreste Parlatano, Mr. Pascal Brandt, equipa MOASIS e Jembi.

Histórico de Revisões				
Versão	**Revisão**	**Data**	**Descrição**	**Autor**
	0.01	2014-01-27	Criação do documento.	Gulam Khan
	0.02	2014-01-28	Revisão do documento	Paulo Grácio

Índice

1 Introdução

1.1 OBJECTIVO

O presente documento tem como objectivo fornecer as informações para instalação e configuração das ferramentas necessárias para o funcionamento do SISMA.

O procedimento de instalação é descrito num simples passo a passo básico e aponta para a facilidade e rapidez.

1.2 ÂMBITO

O âmbito deste documento é consolidar a informação e passos usados necessários para a instalação e configuração do SISMA em um servidor central.

1.3 AUDIÊNCIA

A audiência deste documento é constituída pela entidade contratante Jembi/MOASIS responsável pela especificação e validação do sistema, bem como pela equipa de projecto que venha a ser designada para participar no projecto e pelos elementos da Critical Software que serão responsáveis pelos trabalhos a desenvolver.

1.4 DEFINIÇÕES E ACRÓNIMOS

Acrónimo	Descrição
CSW	Critical Software, SA
MISAU	Ministério da saúde
SGBD	Sistema de Gestão de Base de Dados
SISMA	Sistema de Informação de Saúde para Monitoria e Avaliação
TBC	A confirmar

Tabela 1: Acrónimos

2 Instalação do SISMA

O capítulo de instalação fornece os passos necessários para a instalação do SISMA em um servidor central.

O SISMA funciona em todas as plataformas que possuem uma versão do Java Runtime Environment 6 ou superior, onde inclui os sistemas operativos mais populares como o windows, Linux e Mac. O SIS-MA também funciona em diferentes tipos de base de dados relacionais como o postgreSQL, MySQL, H2 e Derby, e é empacotado como um standard de arquivo Java Web (WAR file), portanto funciona em todos os recipientes servlet, tais como o Tomcat e o Jetty.

O ambiente preferencial para a instalação do SISMA em um servidor é o uso do Sistema operativo recomendado Linux, versão **Ubuntu 12.04 LTS**, usando o motor de base de dados **PostgreSQL** e recepiente Servlet **Tomcat.**

Este documento serve apenas de guia para a instalação do SISMA no ambiente acima mencionado. Para informações mais avançadas deve-se ler a sua documentação oficial.[1]

---

[1] http://www.ubuntu.com/

http://www.postgresql.org/

http://tomcat.apache.org/

2.1 PRÉ-REQUISITOS DO SERVIDOR

O SISMA é uma aplicação com uma base de dados intensiva e requer que o servidor tenha memória RAM, núcleos de CPU e disco rígido adequados. Quanto mais potente o servidor for nas características anteriormente especificadas, melhor a aplicação funcionará.

- Memória RAM: Pelo menos 4 GB por uma instância pequena, 12 GB para uma instância média.
- Núcleo de CPU: 4 Núcleos de CPU para instância pequena e 8 para média ou larga instância.
- Disco: Idealmente use o disco SSD, caso contrário use um disco de 7200 rpm.

  Velocidade mínima de leitura de 150 Mb/s, 200 Mb/s é bom. 350 Mb/s ou superior é o ideal.

2.2 CONFIGURAÇÃO DO SERVIDOR

Esta secção descreve como configurar uma instância SISMA em um sistema operativo Linux Ubuntu 12.04 de 64 Bits, usando postgreSQL como sistema de base de dados e Tomcat como recepiente Servlet.

O termo “invocar” refere-se a execução de comandos em um terminal .

Para um servidor central, a especificação recomendada é:

- Processador quad-core 2Ghz ou superior;
- Memória RAM de 12 GB ou superior;

Note que é necessário o uso do sistema operativo de 64 bits para o uso de mais de 4 GB de memória RAM.

Para este manual assume-se que 8 GB de RAM é alocado para PostgreSQL e 8 GB de RAM é alocado para Tomcat/JVM e é usado um sistema operativo de 64 bits. Se for a usar configurações diferentes, ajuste de acordo com os seus valores.

É Recomendado que a memória disponível seje dividida mais ou menos igualmente entre o banco de dados e o JVM. Lembre-se de deixar um pouco da memória física para o sistema operativo para o desempenho das suas tarefas, por exemplo, em torno de 2 GB.

Os passos marcados como “opcional”, assim como o passo para o ajuste de desempenho, podem ser efectuados em uma fase posterior.

Criação de um novo utilizador (Opcional)

Não é recomendada a execução usando o utilizador “root”. Se pretender criar um novo utilizador para a execução do SISMA, invoque os seguintes comandos:

Criar um novo utilizador chamado “sisma”:

```
useradd –d /home/sisma –m sisma –s /bin/bash
```

Se não existe um grupo admin criado, este deve ser criado da seguinte forma:

groupadd admin

Atribuir ao utlizador a execução temporária de operações como root:

usermod –G admin sisma

Atribuir uma palavra passe para a conta:

Passwd sisma

Certifique que introduza uma palavra passe forte. Se pretender desabilitar o login remoto para a conta root invoque o comando:

sudo passwd –l root

Ajuste do Sistema operativo Kernel

Estas configurações são opcionais, excepto a configuração da memória partilhada, pois é necessário para a alocação em postgreSQL.

Para abrir o ficheiro de configurações Kernel invoque:

sudo nano /etc/sysctl.conf

No final do ficheiro adicione as seguintes linhas:

kernel.shmmax = 4294967296

net.core.rmem_max = 8388608

net.core.wmem_max = 8388608

Faça com que as alterações tenham efeito invocando:

sudo sysctl -p

Instalar JAVA

Para instalar o Java invoque o seguinte commando:

sudo apt-get install openjdk-7-jdk

Verifique se a instalação correu bem invoncado:

java –version

Instalar a base de dados PostgreSQL

Para a instalação do motor de base de dados PostgreSQL versão 9.2, primeiramente deve-se adicionar o repositório de pacotes Ubuntu para o sistema em uso.

sudo apt-get install python-software-properties

sudo add-apt-repository ppa:pitti/postgresql

sudo apt-get update

Invoque o seguinte comando para instalar o PostgreSQL

sudo apt-get install postgresql-9.2

Mude para o utilizador postgres usando:

sudo su postgres

Crie um utilizador não previlegiado chamado "sisma":

createuser -SDRP sisma

Introduza uma palavra passe segura no prompt.

Crie uma base de dados invocando:

createdb -O sisma sisma2

Retorne a sessão executando:

exit

Após execução dos comandos acima citados, o motor de base de dados PostgreSQL estará instalado, com um utilizador postgreSQL "sisma" e uma base de dados "sisma2".

Faça o ajuste de desempenho, abrindo o seguinte arquivo "postgresql.conf":

sudo nano /etc/postgresql/9.2/main/postgresql.conf

e defina as seguintes propriedades:

shared_buffers = 2400MB

Determine quanta memória PostgreSQL pode usar para o cache de dados. Esta configuração controla como o tamanho da memória compartilhada do kernel deve ser reservada para o PostgreSQL. Deve ser ajustado para cerca de 30 % do total da memória designada para o PostgreSQL.

work_mem = 20MB

Determine a quantidade de memória usada para para ordenação interna e operações hash. Esta configuração é por conexão, por consulta para que uma quantidade de memória possa ser consumida se elevar demasiado.

maintenance_work_mem = 256MB

Determine a quantidade de memória PostgreSQL possa ser utilizada para operações de manutenção como criar índices, correr o vaccum, adicionar chaves estrangeiras. Aumentar este valor pode melhorar o desempenho da criação de índices durante a análise e processos de geração de data mart.

effective_cache_size = 8000MB

Uma estimativa da quantidade de memória é disponibilizada para cache de disco pelo sistema operativo (não uma alocação) e é usada pelo PostgreSQL para determinar se um plano de consulta vai caber na memória ou não. Defini-lo para um valor maior do que o que está disponível irá resultar num mau desempenho. Este valor deve ser inclusivo da definição do *shared_buffers.*O PostgreSQL tem 2 camadas de cache: a primeira camada utiliza a memória partilhada kernel e é controlado pelo configuração do *shared_buffers.* O PostgreSQL delega a segunda camada para o cache de disco do sistema operativo e o tamanho da memória disponível pode ser dado com a configuração do *effective_cache_size.*

checkpoint_segments = 32

PostgreSQL escreve novas operações em um arquivo de log chamado *"WAL segments"* que tem um tamanho de 16 MB.

Quando um número de segmentos é escrito, um checkpoint ocorre. Definir este número para um número maior irá melhorar o desempenho de escrita.

checkpoint_completion_target = 0.8

Determine a percentagem da conclusão do segment antes que o checkpoint ocorra. Defini-lo para um valor maior irá difundir as gravações para fora e reduzir a sobrecarga média de gravação.

wal_buffers = 16MB

Defina a memória usada para o buffer durante o processo de gravação WAL. Aumentar este valor pode melhorar o rendimento em sistemas de gravação pesada.

synchronous_commit = off

Especifique se a transação for confirmada vai esperar por os registros WAL serem gravados no disco antes de retornar para o cliente ou não. Configurando-o para "*off*" irá melhorar consideravelmente o desempenho. Implica também que há um ligeiro atraso do relatório de transação com sucesso para o cliente, mas o estado do banco de dados não pode ser corrompido e esta é uma boa alternativa para o desempenho intensivo e escrita em sistema pesados como o SIS-MA.

wal_writer_delay = 10000ms

Especifique o atraso entre as operações de escrita WAL. Defini-lo para um valor maior irá aumentar o desempenho em sistema de escrita em massa, uma vez que potencialmente muitas operações de escrita podem ser executadas em uma única descarga para o disco.

Faça o restart do PostgreSQL invocando:

sudo /etc/init.d/postgresql restart

Definir a configuração da base de dados

A informação da conexão a base de dados é fornecida no SISMA atráves da configuração do ficheiro chamado "hibernate.properties".

Crie esse ficheiro e guarde é um local conveniente. Um ficheiro correspondente para as configurações acima tem as seguintes propriedades:

hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect

hibernate.connection.driver_class = org.postgresql.Driver

hibernate.connection.url = jdbc:postgresql:sisma2

hibernate.connection.username = sisma

hibernate.connection.password = xxxx

hibernate.hbm2ddl.auto = update

Um erro comum é ter um espaço em branco após o último valor da propriedade. Certifique-se que não há espaço em branco no final de qualquer linha. Lembre-se também que este ficheiro contém a senha em texto claro para o seu banco de dados sisma2, logo, é preciso proteger contra acessos não autorizados. Para isso invoque:

chmod invoke 0600 hibernate.properties

Garantindo assim que somente o utilizador sisma tem permissões de leitura e escrita sobre o ficheiro.

Instalar o Tomcat

Faça Download do Tomcat apartir do link:

http://tomcat.apache.org/download-70.cgi

Uma ferramenta útil para download de ficheiros da web é o *"wget"*.

Extrai para uma localização conveniente. Este manual pressupõe que se tenha navegado para o diretório root do arquivo extraído.

Limpe todas as aplicações web pré-instaladas invocando:

rm -rf webapps/*

Apartir da pasta do projecto SISMA, na dropbox, \SISMA - Management Repository\deliverables\Phase 4 - Software development, copie os ficheiros **"SISMA.WAR"** e "**ROOT.war**" para o directório *webapps* do ambiente.

Crie um ficheiro bin/setenv.sh e adicione as linhas abaixo. A primeira vai definir a localização do seu Java Runtime Environment, o segundo vai dedicar memória para o Tomcat e o terceiro irá definir o local para onde SIS-MA vai procurar o ficheiro de configuração hibernate.properties.

Por favor verifique se o caminho dos binários do Java estão corretas, já que podem variar de sistema para sistema, por exemplo, em sistemas AMD, pode-se ver em / java-7-openjdk-amd64. Note que deverá ajustar esta ao seu ambiente:

export JAVA_HOME='/usr/lib/jvm/java-7-openjdk'

export JAVA_OPTS='-Xmx7500m -Xms4000m -XX:MaxPermSize=500m -XX:PermSize=300m'

export SIS-MA_HOME='/home/sisma/config'

Se for preciso alterar a porta que o Tomcat atende a pedidos, abra o ficheiro de configuração do Tomcat /conf/server.xml, localize o elemento <Connector> que não é comentada e altere o valor do atributo da porta para o número da porta desejada.

Para monitorar o comportamento do Tomcat o log é a principal fonte de informação. Os registros podem ser facilmente visualizados a partir do comando:

tail -f logs/catalina.out

Definir o tempo do servidor

Deve-se definir correctamente a data do servidor de acordo com o fuso horário em que a parte principal dos utilizadores do sistema estão localizados. Isso vai afectar, por exemplo, quando as tarefas agendadas são executadas.

Executar o SISMA

Faça o script do executável de inicialização invocando:

chmod 755 bin/*

O SISMA pode ser iniciado/interrompido invocando:

bin/startup.sh

bin/shutdown.sh

O ficheiro log pode ser monitorado através do comando:

tail -f logs/catalina.out

Para aceder ao instância SISMA, abra um browser preferencialmente o google chrome e coloque o seguinte link:

http://localhost:8080

Para aceder a partir de um outro computador na rede, use: http://IP do servidor:8080